This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1. (Currently Amended) A system that manages the partitioning of an application comprising:

at least one processor and at least one memory in communication with said at least one processor, said processor configured to execute program instructions that comprise the following:

a base component stored in said at least one memory that hosts [[the]] an operation of a first environment and a second environment, the application comprising:

a first software object of said application that executes in said first environment comprising a first operating system, wherein said first software object provides a subset of the operations of the application; said first software object handling a plurality of data and including logic to identify a first of said plurality of data as not processable by said first software object, said first software object sending said first of said plurality of data to said base component ~~environment~~, said first software object receiving processed data corresponding to said first of said plurality of data from said base component ~~environment~~, said first software object using said processed data to further process the plurality of data; and

a second software object of said application that executes in said second environment comprising a second operating system, said second software object receiving said first of said plurality of data from said base component with a corresponding wrapper, said second software object verifying said first of said plurality of data as being unmodified by comparing said data to the corresponding wrapper, said second software object processing said first of said plurality of data in a manner that resists tampering with said first of said plurality of data, said second software object sending said processed data to said base component;

said base component comprising or hosting logic that receives said first of said plurality of data from said first software object, applies the corresponding wrapper to said first of said plurality of data, said corresponding wrapper comprising an indication of said second software object and a seal that may be checked against said first of said plurality of data to determine whether said first of said plurality of data has been altered since the seal

was determined, and routes said first of said plurality of data to said second environment, such that functionality of said application is parsed between said first and second operating systems;

said base component further comprising or hosting logic that receives said processed data from said second software object, applies a second wrapper to said processed data, said second wrapper comprising an indication of said first software object and a second seal that may be checked against said processed data to determine whether said processed data has been altered since the second seal was determined, and routes said processed data to said first environment, such that functionality of said application is parsed between said first and second operating systems.

2. (Previously Presented) The system of claim 1, wherein said first software object causes a representation of said first of said plurality of data to be displayed on a display device, said representation comprising one or more indecipherable graphics.

3. (Currently Amended) The system of claim 2, wherein said one or more indecipherable graphics are either: [[(1)]] the same size as each other, or [[(2)]] of sizes that are unrelated to [[the]] a content of said first of said plurality of data.

4. (Currently Amended) The system of claim 1, and wherein the resistance to tampering provided by said second software object comprises said second environment resisting interference with [[the]] a display of said first of said plurality of data by writing a representation of said first of said plurality of data into a video memory associated with a display device so as to cause said representation to supersede any image at a location on said display device at which said representation is to be displayed.

5. (Currently Amended) The system of claim 1, wherein said first of said plurality of said is entered on a keyboard, and wherein the ~~resistant~~ resistance to tampering provided by said second software object comprises resisting tampering with said first of said plurality of data in transit from said keyboard to an input stream of said second software object.

6. (Currently Amended) The system of claim 5, wherein said second software object wraps [[signs]] said first of said plurality of data to prevent subsequent tampering with said first of said plurality of data.

7. (Currently Amended) The system of claim 6, wherein said second environment wraps [[signs]] said first of said plurality of data with a wrapper and the ~~signature created by said second application as~~ wrapper is an indication that said first of said plurality of data and said signature were created in said second environment.

8. (Previously Presented) The system of claim 1, wherein said base component comprises a component that assigns a first identifier to said second environment.

9. (Original) The system of claim 8, wherein said first of said plurality of data includes, or is accompanied by, said first identifier and a second identifier that identifies said second software object.

10. (Currently Amended) The system of claim 1, wherein said first environment is associated with a first specification that describes [[the]] a behavior of said first environment, wherein said second environment is associated with a second specification that describes the behavior of said second environment, wherein there is a higher level of assurance that said second environment will conform to said second specification than that said first environment will conform to said first specification.

11. (Original) The system of claim 10, wherein said second software object relies upon the behavior of the second environment in order to resist tampering with said first of said plurality of data.

12. (Previously Presented) The system of claim 1, wherein said base component is said second environment, or is included within said second environment.

13. (Currently Amended) A method of a first software object of an application, which executes in a first environment comprising a first operating system executing on a computer, handling data to which an assurance policy that corresponds to a level of assurance

that the application will perform its expected functions correctly applies, the method comprising:

determining, by the first software object, that the data should be processed securely;

sending, by the first software object, the data to a base environment with an indication to process the data securely;

determining, by the base environment, a second software object with which to process the data securely, the second software object corresponding to the first software object;

applying, by the base environment, a wrapper to the data, the wrapper identifying [[the]] a second software environment of the second software object, the wrapper comprising a seal that may be checked against the data to determine whether the data has been altered since the seal was determined;

sending, by the base environment, the data and the corresponding wrapper to the second software environment;

creating, by the second software environment, a resistance to tampering comprising determining that the data has not been modified using the data and the corresponding seal;

processing, by the second software environment, the data;

sending, by the second software environment, [[the]] a result of the processed data to the base environment with an indication to return the data to a software environment that originally sent the data;

determining, by the base environment, that the first software environment is the software environment that originally sent the result;

applying, by the base environment, a second wrapper to the result, the second wrapper identifying the first software environment, the second wrapper comprising a second seal that may be checked against the result to determine whether the result has been altered since the second seal was determined;

sending, by the base environment, the result and the second wrapper to the first software environment; and

verifying, by the first software environment, that the result has not been modified using the result and the second seal.

14. (Original) The method of claim 13, wherein the resistance to tampering comprises a resistance to a change in said data.

15. (Original) The method of claim 14, wherein said data is to be displayed on a visual display device, and wherein the resistance to tampering comprises displaying a representation of said data in a location on said visual display device and superseding any image other than said representation that is rendered at said location.

16. (Previously Presented) The method of claim 13, further comprising:

directing, by said first software object a component responsible for visual output to display a representation of the data to be displayed on a visual display device, said representation comprising one or more indecipherable graphics.

17. (Currently Amended) The method of claim 16, wherein said representation are either: [[(1)]] the same size as each other, or [[(2)]] of sizes that are unrelated to [[the]] a content of said data.

18. (Currently Amended) The method of claim 16, further comprising:

directing, by said first software object or said second software object, or a combination of said first software object and said second software object, a component responsible for visual output to display items displayed on said visual display device to be changed in at least one respect to allow permit viewing of an image of the data produced by said second software object.

19. (Currently Amended) The method of claim 14, wherein said data is provided using a keyboard, and wherein the resistance to tampering comprises resisting a change to the data in transit from the keyboard to [[the]] an input stream of the second software object.

20. (Currently Amended) The method of claim 13, wherein said assurance policy specifies that said data is to be handled by said second software object.

21. (Original) The method of claim 13, wherein said data includes, or is associated with, a first label that identifies said second environment as a location in which said data is to be processed.

22. (Original) The method of claim 21, wherein said data includes, or is associated with, a second label that identifies said second software object as a processor for said data, and wherein said second environment routes said data to said second software object based on said second label.

23. (Currently Amended) The method of claim 13, wherein said second environment is associated with a first specification that describes [[the]] a behavior of said second environment, and wherein said assurance policy provides that said second environment will conform to said first specification.

24. (Currently Amended) The method of claim 13, wherein said first environment is associated with a second specification that describes [[the]] a behavior of said first environment, and wherein said first environment provides a second level of assurance that actions performed in the first environment will be performed correctly, said second level of assurance being relatively lower than said first level of assurance.

25. (Currently Amended) A computer-readable storage medium having stored thereon code and data to allow a user to operate on first and second types of data, said second type of data requiring a relatively higher level of protection from tampering than said first type of data, said code and data comprising:

determining, by [[the]] a first software object of a first software environment, that the data should be processed securely;

sending, by the first software object, the data to a base environment with an indication to process the data securely;

determining, by the base environment, a second software object with which to process the data securely, the second software object corresponding to the first software object;

applying, by the base environment, a wrapper to the data, the wrapper identifying [[the]] a second software environment of the second software object, the wrapper comprising a seal that may be checked against the data to determine whether the data has been altered since the seal was determined;

sending, by the base environment, the data and the corresponding wrapper to the second software environment;

verifying, by the second software environment, that the data has not been modified using the data and the corresponding seal;

processing, by the second software environment, the data;

sending, by the second software environment, [[the]] a result of the processed data to the base environment with an indication to return the data to a software environment that originally sent the data;

determining, by the base environment, that the first software environment is the software environment that originally sent the result;

applying, by the base environment, a second wrapper to the result, the second wrapper identifying the first software environment, the second wrapper comprising a second seal that may be checked against the result to determine whether the result has been altered since the second seal was determined;

sending, by the base environment, the result and the second wrapper to the first software environment;

verifying, by the first software environment, that the result has not been modified using the result and the second seal;

determining, by the first software environment, that the second ~~computing~~ software environment has a corresponding level of trust sufficient for the first ~~computing~~ software environment to trust the result; and

using the result to execute the first software object.

26. – 34. (Cancelled)

35. (Previously Presented) The method of claim 13, further comprising:

determining, by the first software environment, that the second software environment has a corresponding level of trust sufficient for the first software environment to trust the result; and

using the result to execute the first software object.